

WIRTSCHAFTSINFORMATIK 1

CODIERUNG

PROF. DR. CHRISTIAN BOCKERMANN, PROF. DR. JULIAN GROTE, PROF. DR.
VOLKER KLINGSPOR

HOCHSCHULE BOCHUM

WINTERSEMESTER 2026/27

Inhalt

- 1 Hardware
- 2 Software
- 3 Speicherorte von Dateien
- 4 Codierung von Daten
- 5 Fehler erkennen
- 6 Codierung von Bildern
- 7 Codierung von Ton
- 8 Komprimierung
- 9 Zusammenfassung

HARDWARE

Welchen Computer sollte ich mir kaufen?

- Bauform (Form Factor)
- Prozessor (CPU)
- Kerne
- Hauptspeicher / RAM
- Festplatte / SSD

Welchen Computer sollte ich mir kaufen?

- Bauform (Form Factor)
- Prozessor (CPU)
- Kerne
- Hauptspeicher / RAM
- Festplatte / SSD

Schauen Sie nach: Wie viel RAM und wie viel Festspeicher hat Ihr Laptop oder Smartphone?

Die Bauform beeinflusst den Preis, die Leistungsfähigkeit und die Erweiterungsfähigkeit

- Laptop (Dockingstation)
- Desktop (unterschiedliche Varianten)
- Tablet / Smartphone
- Ein-Platinen-Computer

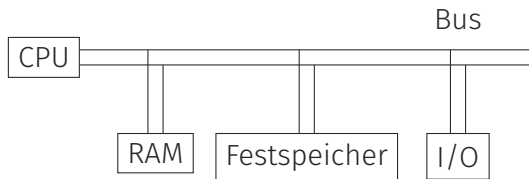


CPU – Central Processing Unit (Prozessor)



- Führt Anweisungen von Programmen aus
- Eigenschaften:
 - Takt (GHz):** Anzahl der Befehle pro Sekunde
 - Kerne (Core):** Anzahl der gleichzeitig ausführbaren Befehle
 - Cache (MB):** Zwischenspeicher
 - GPU:** Grafikprozessor: viele einfache Kerne parallel
wichtig für Grafik, Videobearbeitung und KI

Arbeitsspeicher / Hauptspeicher (RAM)



- Sehr *schneller* Speicher für Daten
- Speichert Daten und Programme **während des Betriebs** des Computers
- Eigenschaften:

Größe (GB): Insbesondere bei Grafik- und Videobearbeitung wichtig

Preis: Im Vergleich zu Festspeicher sehr teuer!

Festspeicher / Festplatte

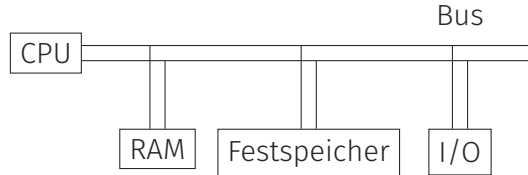


- Festplatte, SSD, NVME
- Speichert Daten und Programme **dauerhaft**, wiederbeschreibbar
- Eigenschaften:

Typ (Festplatte (HD) / SSD)

Größe (GB/TB)

Input/Output – I/O (Eingabe- und Ausgabe)



- Sämtliche Ein- und Ausgabegeräte (Peripherie-Geräte)
- Eigenschaften:

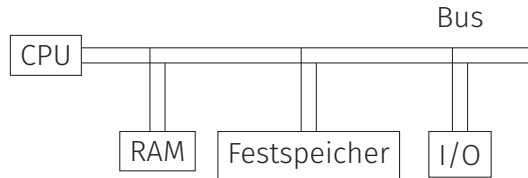
Monitor: Auflösung und Größe

HDMI: Schnittstelle für Bild und Ton

USB A: Schnittstelle für externe Geräte (Drucker, Tastatur, ...)

USB C: Schnittstelle für externe Geräte (Drucker, Tastatur, Video, Ton, Netzwerk, Strom, ...)

Input/Output – I/O (Eingabe- und Ausgabe)



- Sämtliche Ein- und Ausgabegeräte (Peripherie-Geräte)

*In den Poolräumen können Sie USB-C nutzen,
um Monitor/Tastatur/Maus an Ihr Laptop anzuschließen
und es gleichzeitig zu laden.*

SOFTWARE

Programme, die auf der Hardware ausgeführt werden.

- Folge von Anweisungen
- Auf dem Festspeicher installiert
- Zur Ausführung in den Arbeitsspeicher (RAM) geladen

Betriebssystem

MS Windows, MacOS, Linux, Android, iOS

Anwendungssoftware

Tabellenkalkulation, Browser, Email-Client, Smartphone-Apps

Software kann lokal oder auf einem Server laufen.

Lokal

- Die Software ist auf Ihrem Computer installiert
- Sie haben die Kontrolle über die Ausführung (Starten, Beenden, Computer ausschalten)
- Die Daten können lokal oder auf einem Server gespeichert werden.

Tabellenkalkulation, Textverarbeitung, ...

MS Excel, MS Word, Thunderbird, ...

Software kann lokal oder auf einem Server laufen.

Server / Cloud

- Die Software wird von einem Unternehmen auf einem Server installiert
- Das Unternehmen hat die Kontrolle über die Ausführung
- Die Daten werden auf dem Server gespeichert.

MS Office 365

Webmail

Social Media Dienste

Web-Anwendung (im Browser) *und* lokale Anwendung (App)
mit Zugriff auf serverseitig gespeicherte Daten

- Email
- Maps
- Social Media Dienste

TIPP!

Verwenden Sie für die Hochschul-Emails einen Email-Client auf Ihrem Smartphone, damit Sie keine wichtigen Emails verpassen.

- Sie können eine eigene App dafür verwenden, um private Emails davon zu trennen.
- Sie richten lokal ein Email-Konto ein, mit dem Sie auf die Hochschul-Email zugreifen können.
- Anleitung: **https:**
[//www.hochschule-bochum.de/cit/anleitungen/studmail-1/](https://www.hochschule-bochum.de/cit/anleitungen/studmail-1/)

SPEICHERORTE VON DATEIEN

- Lokal auf dem eigenen Computer
- Netzlaufwerke
- Cloud-Dienste

Dateien speichern auf der Festplatte

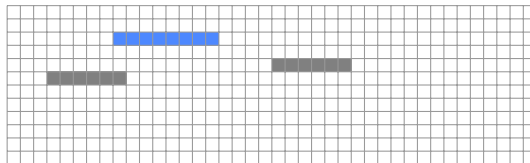
Festplatten speichern Bits z.B.

- durch Magnetisierung von winzigen Stellen (Zellen) auf Scheiben
- oder in Zellen von nicht-flüchtigen Chips (SSD)
- jede Zelle hat eine Position/Adresse

Dateien speichern auf der Festplatte

Festplatten speichern Bits z.B.

- durch Magnetisierung von winzigen Stellen (Zellen) auf Scheiben
- oder in Zellen von nicht-flüchtigen Chips (SSD)
- jede Zelle hat eine Position/Adresse



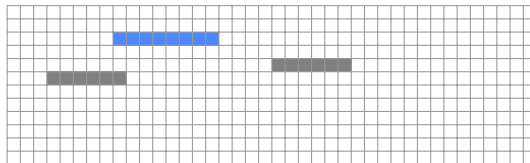
■ Datei Dokument1.docx

■ Datei Tabelle.xlsx

Dateien speichern auf der Festplatte

Festplatten speichern Bits z.B.

- durch Magnetisierung von winzigen Stellen (Zellen) auf Scheiben
- oder in Zellen von nicht-flüchtigen Chips (SSD)
- jede Zelle hat eine Position/Adresse



■ Datei Dokument1.docx

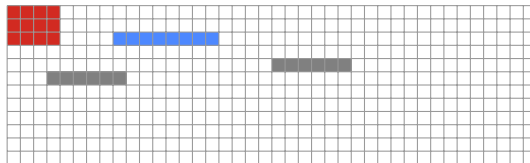
■ Datei Tabelle.xlsx

Woher wissen wir, wo die Bits von Dokument1.docx liegen?

Dateien speichern auf der Festplatte

Festplatten speichern Bits z.B.

- durch Magnetisierung von winzigen Stellen (Zellen) auf Scheiben
- oder in Zellen von nicht-flüchtigen Chips (SSD)
- jede Zelle hat eine Position/Adresse



- Datei `Dokument1.docx`
- Datei `Tabelle.xlsx`
- Inhaltsverzeichnis (FAT)

Dateien speichern auf der Festplatte

Das Dateisystem verwaltet, wo die Bits einer Datei gespeichert sind und wer auf welche Dateien zugreifen kann.

File Allocation Table (FAT):

Datei	Start	Ende
C:\Dok1.docx	392	481
C:\Tabelle.xlsx	879	1537

Wie finde ich Dateien auf meinem Computer?

- Jeder Benutzer hat ein eigenes (Wurzel-) Verzeichnis
MacOS: /Users/Benutzername
Windows: C:\Users\Benutzername
- Verzeichnisse (Ordner) können ineinander liegen
- Ein *Pfad* verweist auf den genauen Ort der Datei
 - /Users/Benutzername/Desktop/bild.jpg
 - /Users/Benutzername/Documents/buero/abrechnung.doc
- Pfade können relativ sein
../../Desktop/bild.jpg
- ein Punkt in einem Pfad verweist auf das aktuelle Verzeichnis
- zwei Punkte in einem Pfad verweisen auf das übergeordnete Verzeichnis

- Es können weitere Speichermedien lokal eingebunden werden (auch über USB)
- Bei Defekt des Speichermediums oder Diebstahl sind die Dateien verloren
→ an Sicherung der Dateien denken
- Ransomware hat Zugriff auf die Dateien und kann sie verschlüsseln
<https://de.wikipedia.org/wiki/Ransomware>
→ an Sicherung der Dateien denken

- Netzlaufwerke sind Verzeichnisse, die auf einem anderen Computer liegen
- Sie werden als Laufwerke im eigenen Computer eingebunden
Protokolle: NFS, Samba (SMB)
- Sie funktionieren genauso wie lokale Verzeichnisse
- Mehrere Benutzer können sich die Laufwerke teilen
- Sie werden in Unternehmen üblicherweise automatisch gesichert

- Dateien liegen auf Web-Servern im Internet
- GoogleDrive, MS OneDrive, Apple iCloud, **Sciebo**
- Dateien müssen für Bearbeitung mit lokalen Programmen heruntergeladen werden
Explizit: Beim Download wird das Zielverzeichnis festgelegt
Implizit: Die Datei wird in einem Verzeichnis für temporäre Dateien gespeichert
- Betriebssystem oder spezielle Software synchronisiert lokale Verzeichnisse mit Cloud-Verzeichnissen
- Dienstleister sorgt für Sicherung der Daten
- Unternehmen müssen ggf. sicherstellen, dass die Daten innerhalb der EU gespeichert werden

CODIERUNG VON DATEN

Wie können wir Informationen im Computer speichern?

- Zahlen
- Texte
- Fotos
- Ton

Digitalisierung: Übertragung von analogen Informationen in digitale Daten!

- Binärsystem
- 0 und 1 werden als einzige Ziffern verwendet
- Einfach im Computer speicherbar
Spannung liegt an $\longleftrightarrow$ liegt nicht an

Natürliche Zahlen

- Folge von Ziffern
- Jede Ziffer hat abhängig von der Stelle eine Wertigkeit

Natürliche Zahlen

- Folge von Ziffern
- Jede Ziffer hat abhängig von der Stelle eine Wertigkeit
- Dezimalsystem
 - Wert jeder Stelle wächst um Faktor 10
 $341 = 3 * 100 + 4 * 10 + 1 * 1$
 - Erfordert 10 Ziffern
→ Schwierig mit unterschiedlichen Spannungen im Computer darzustellen

Natürliche Zahlen

- Folge von Ziffern
- Jede Ziffer hat abhängig von der Stelle eine Wertigkeit
- Dezimalsystem
 - Wert jeder Stelle wächst um Faktor 10
 $341 = 3 * 100 + 4 * 10 + 1 * 1$
 - Erfordert 10 Ziffern
→ Schwierig mit unterschiedlichen Spannungen im Computer darzustellen
- Binärsystem
 - Wert jeder Stelle wächst um Faktor 2
 $1101 = 1 * 8 + 1 * 4 + 0 * 2 + 1 * 1$
 - Erfordert 2 Ziffern

- Wieviel Stellen benötigen Binärzahlen?
 - $2^{3.3} = 10 \iff \log_2(10) = 3.3$
 - Es werden 3,3 Binärstellen benötigt, um 10 Zahlen zu speichern
 - Binärzahlen sind also ca. 3,3 mal so lang wie Dezimalzahlen
- Größe des Speicherplatzes limitiert die Größe der Zahlen
 - Z.B. für 2727331 benötigen wir $\log_2(2727331) = 21,37$ Stellen
 - 2 Byte (16 Bit): $2^{16} = 65536$
 - 4 Byte (32 Bit): $2^{32} = 4294967296$

Welche Zahl versteckt sich hinter 101101?

Welche Zahl versteckt sich hinter 101101?

Stellenwert	32	16	8	4	2	1
Ziffer	1	0	1	1	0	1

Welche Zahl versteckt sich hinter 101101?

Stellenwert	32	16	8	4	2	1
Ziffer	1	0	1	1	0	1

$$32 + 8 + 4 + 1 = 45$$

- Merksatz: die Stellenwerte der Einsen addieren

Und wie kommt man von 45 zurück zu 101101?

Halbieren mit Rest:

$$\begin{array}{rcl} 45 : 2 = & 22 & \text{Rest } 1 \\ 22 : 2 = & 11 & \text{Rest } 0 \\ 11 : 2 = & 5 & \text{Rest } 1 \\ 5 : 2 = & 2 & \text{Rest } 1 \\ 2 : 2 = & 1 & \text{Rest } 0 \\ 1 : 2 = & 0 & \text{Rest } 1 \end{array}$$

Und wie kommt man von 45 zurück zu 101101?

Halbieren mit Rest:

$$\begin{array}{rcl} 45 : 2 = & 22 & \text{Rest } 1 \\ 22 : 2 = & 11 & \text{Rest } 0 \\ 11 : 2 = & 5 & \text{Rest } 1 \\ 5 : 2 = & 2 & \text{Rest } 1 \\ 2 : 2 = & 1 & \text{Rest } 0 \\ 1 : 2 = & 0 & \text{Rest } 1 \end{array}$$

- Die Reste von unten nach oben lesen: 101101
- Kontrolle: Stellenwerte der Einsen addieren → 45

Rechnen Sie Ihr Geburtsjahr in eine Binärzahl um!

- Verfahren: fortgesetzt halbieren, Reste notieren
- Tipp: es werden 11 Binärstellen
- Für Schnelle: das Alter geht mit 5–6 Stellen

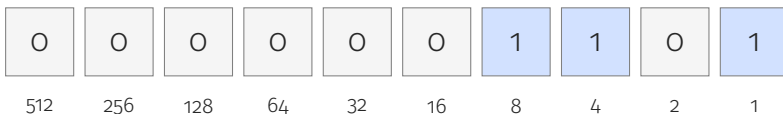
Rechnen Sie Ihr Geburtsjahr in eine Binärzahl um!

- Verfahren: fortgesetzt halbieren, Reste notieren
- Tipp: es werden 11 Binärstellen
- Für Schnelle: das Alter geht mit 5–6 Stellen
- Probe: Stellenwerte der Einsen addieren — kommt Ihr Geburtsjahr heraus?

Bis wie viel können Sie mit 10 Fingern zählen?

Bis wie viel können Sie mit 10 Fingern zählen?

- Nicht bis 10 — bis **1023**!
- Jeder Finger ist ein Bit: gestreckt = 1, gebeugt = 0



$$8 + 4 + 1 = 13$$

111111110000000011001100

Können Sie sich das merken? Vorlesen? Fehlerfrei abtippen?

111111110000000011001100

Können Sie sich das merken? Vorlesen? Fehlerfrei abtippen?

- Idee: 4er-Gruppen bilden, jede Gruppe bekommt **ein** Zeichen

1111	1111	0000	0000	1100	1100
F	F	0	0	C	C

- Stellenwertsystem mit Basis 16
- 16 Ziffern: 0–9 und A–F
- Eine Hex-Ziffer = vier Bit
- Nur eine kompakte **Schreibweise** für Menschen
 - der Computer speichert weiter Bits

- Stellenwertsystem mit Basis 16
- 16 Ziffern: 0–9 und A–F
- Eine Hex-Ziffer = vier Bit
- Nur eine kompakte **Schreibweise** für Menschen
 - der Computer speichert weiter Bits
- Rechnen muss hier niemand:
nur 4er-Gruppen übersetzen

Binär	Hex	Binär	Hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Was bedeutet #FF0000?

Was bedeutet #FF0000?



#FF0000



#00FF00



#0000FF



#FFFFFF



#000000

- Drei Byte: Rot, Grün, Blau — je zwei Hex-Ziffern
- FF = 255 = volle Stärke, 00 = aus
- Bekannt aus Webdesign, PowerPoint, Instagram ...
- Warum genau drei Farben? → Kapitel Bilder

Was bedeutet #FF0000?



#FF0000



#00FF00



#0000FF



#FFFFFF



#000000

- Drei Byte: Rot, Grün, Blau — je zwei Hex-Ziffern
- FF = 255 = volle Stärke, 00 = aus
- Bekannt aus Webdesign, PowerPoint, Instagram ...
- Warum genau drei Farben? → Kapitel Bilder
- Auch sonst überall, wo Bitmuster lesbar sein sollen:
MAC-Adressen, IPv6-Adressen, Fehlercodes, ...

Reelle Zahlen

- Zwischen zwei Zahlen liegen unendlich viele andere Zahlen
- Können nicht einfach "aufgereiht" werden
- Speichern einer natürlichen Zahl, die als 0,... interpretiert wird
- Speichern eines Exponenten, der die Verschiebung dieser Zahl um entsprechend viele Stellen angibt.
- Beispiel: $123.456 \iff 0.123456 * 1000$
- Gespeichert wird 123456 und 3 Stellen nach links verschieben.
- Zahlen können *sehr* groß bzw. *sehr* klein werden
- Anzahl der gespeicherten Stellen ist immer gleich

Übliche Codierung für reelle Zahlen: IEEE 754

- Einfache Genauigkeit
 - 7 ...8 Stellen (im Sinne des Dezimalsystems) werden gespeichert
 - 32 Bit Platzbedarf für die gesamte Zahl
- Doppelte Genauigkeit
 - 15 ...16 Stellen werden gespeichert
 - 64 Bit Platzbedarf für die gesamte Zahl
- Rundungsfehler bei Berechnungen, wenn die Zahlen unterschiedlich groß sind

Übliche Codierung für reelle Zahlen: IEEE 754

- Einfache Genauigkeit
 - 7 ...8 Stellen (im Sinne des Dezimalsystems) werden gespeichert
 - 32 Bit Platzbedarf für die gesamte Zahl
- Doppelte Genauigkeit
 - 15 ...16 Stellen werden gespeichert
 - 64 Bit Platzbedarf für die gesamte Zahl
- Rundungsfehler bei Berechnungen, wenn die Zahlen unterschiedlich groß sind
- Auch in Excel: $=0,1+0,2=0,3$ ergibt FALSCH
intern wird binär gerundet, das Ergebnis ist nur *fast* 0,3
- Geldbeträge deshalb nie auf Gleichheit prüfen (→ Termin Excel)

Buchstaben

- Jedem Buchstaben/Zeichen wird eine Zahl zugeordnet
 - A: 65
 - B: 66
 - ...
- Ursprünglich (ASCII): 7 Bit: Platz für 128 Zeichen
- Amerikanischer Zeichensatz (keine Umlaute)
- Zunächst unterschiedliche Ergänzungen des 8. Bits bei verschiedenen Betriebssystemen
dadurch verschiedene Codes für Umlaute und Sonderzeichen

Umlaute und Sonderzeichen

Aktuelle Kodierungen:

- ISO 8859: Familie von Kodierungen für verschiedene Sprachräume
- UTF-8: Universelle Kodierung

Kennen Sie das?

Herzliche GrÃ¼Ãe aus MÃ¼nchen

Kennen Sie das?

Herzliche GrÃ¼Ãe aus MÃ¼nchen

- UTF-8 speichert ü als **zwei** Byte
- Liest ein Programm diese Bytes mit einer alten 1-Byte-Tabelle (ISO 8859), werden daraus zwei seltsame Zeichen: Ã¼
- Gleiche Bits — falsche Interpretation!

Kennen Sie das?

Herzliche GrÃ¼Ãe aus MÃ¼nchen

- UTF-8 speichert ü als **zwei** Byte
- Liest ein Programm diese Bytes mit einer alten 1-Byte-Tabelle (ISO 8859), werden daraus zwei seltsame Zeichen: Ã¼
- Gleiche Bits — falsche Interpretation!
- Übrigens: Auch Emojis sind nur Zahlen — das Grinse-Gesicht hat die Unicode-Nummer 128.512

FEHLER ERKENNEN

- Übertragung und Speicherung sind Physik: Störungen kommen vor
- Ein einziges gekipptes Bit kann viel ändern:
 - $1000001 = A \rightarrow 1100001 = a$
 - aus 8 Euro (001000) werden 40 Euro (101000)

Wie merkt der Empfänger, dass etwas kaputt ist?

- Übertragung und Speicherung sind Physik: Störungen kommen vor
- Ein einziges gekipptes Bit kann viel ändern:
 - $1000001 = A \rightarrow 1100001 = a$
 - aus 8 Euro (001000) werden 40 Euro (101000)

Wie merkt der Empfänger, dass etwas kaputt ist?

- Wir spendieren ein zusätzliches **Kontroll-Bit**

Regel: Das Zusatzbit wird so gewählt, dass die Anzahl der Einsen **gerade** ist.

- **1000001** → 2 Einsen → Paritätsbit 0 → **10000010**
- **1100001** → 3 Einsen → Paritätsbit 1 → **11000011**

Der Empfänger zählt nach: ungerade Anzahl Einsen → Fehler!

Regel: Das Zusatzbit wird so gewählt, dass die Anzahl der Einsen **gerade** ist.

- **1000001** → 2 Einsen → Paritätsbit 0 → **10000010**
- **1100001** → 3 Einsen → Paritätsbit 1 → **11000011**

Der Empfänger zählt nach: ungerade Anzahl Einsen → Fehler!

- Erkennt jeden einzelnen Bitfehler
- Erkennt *nicht*, wenn zwei Bits gleichzeitig kippen
- Verrät nicht, *welches* Bit gekippt ist ...

Regel: Das Zusatzbit wird so gewählt, dass die Anzahl der Einsen **gerade** ist.

- **1000001** → 2 Einsen → Paritätsbit 0 → **10000010**
- **1100001** → 3 Einsen → Paritätsbit 1 → **11000011**

Der Empfänger zählt nach: ungerade Anzahl Einsen → Fehler!

- Erkennt jeden einzelnen Bitfehler
- Erkennt *nicht*, wenn zwei Bits gleichzeitig kippen
- Verrät nicht, *welches* Bit gekippt ist ...
- ... oder doch? → Zaubertrick!

0	1	1	0	1	0
1	1	0	1	0	1
0	0	1	1	1	0
1	0	0	1	0	0
1	1	1	0	1	1
0	1	0	0	1	1

- Eine Person füllt das 6×6-Feld beliebig mit 0 und 1

0	1	1	0	1	0	1
1	1	0	1	0	1	0
0	0	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	1	1
0	1	0	0	1	1	1
1	0	1	1	0	1	0

- Eine Person füllt das 6×6-Feld **beliebig** mit 0 und 1
- Ich ergänze eine Zeile und eine Spalte – nach einer geheimen Regel

0	1	1	0	1	0	1
1	1	0	1	0	1	0
0	0	1	0	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	1	1
0	1	0	0	1	1	1
1	0	1	1	0	1	0

- Jemand dreht heimlich ein Feld um —
ich finde es sofort!

0	1	1	0	1	0	1
1	1	0	1	0	1	0
0	0	1	0	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	1	1
0	1	0	0	1	1	1
1	0	1	1	0	1	0

- Jemand dreht heimlich ein Feld um — ich finde es sofort!
- Diese Zeile und diese Spalte haben eine **ungerade** Anzahl Einsen

0	1	1	0	1	0	1
1	1	0	1	0	1	0
0	0	1	0	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	1	1
0	1	0	0	1	1	1
1	0	1	1	0	1	0

- Jemand dreht heimlich ein Feld um — ich finde es sofort!
- Diese Zeile und diese Spalte haben eine **ungerade** Anzahl Einsen
- Der Fehler liegt an der **Kreuzung**

- Die 7. Spalte ist das **Paritätsbit** jeder Zeile
- Die 7. Zeile ist das Paritätsbit jeder Spalte
- Ein gekipptes Feld macht genau eine Zeile *und* eine Spalte ungerade
- Die Kreuzung verrät die Position — das Bit kann sogar **korrigiert** werden

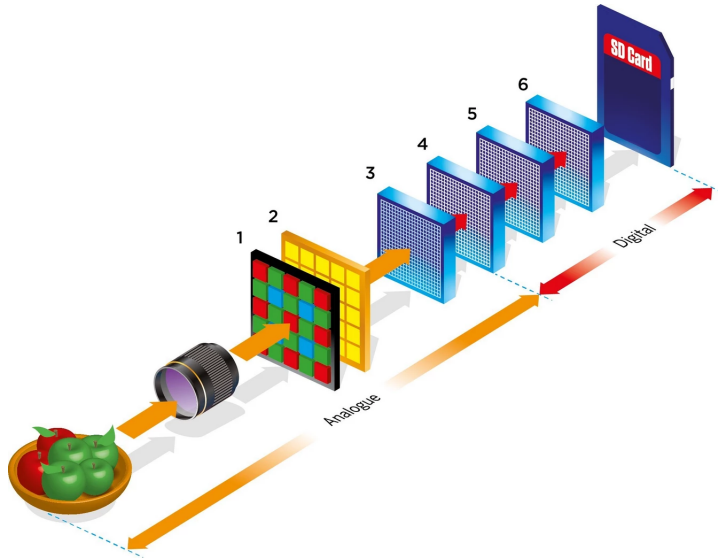
- Die 7. Spalte ist das **Paritätsbit** jeder Zeile
- Die 7. Zeile ist das Paritätsbit jeder Spalte
- Ein gekipptes Feld macht genau eine Zeile *und* eine Spalte ungerade
- Die Kreuzung verrät die Position — das Bit kann sogar **korrigiert** werden

Wo steckt diese Idee überall?

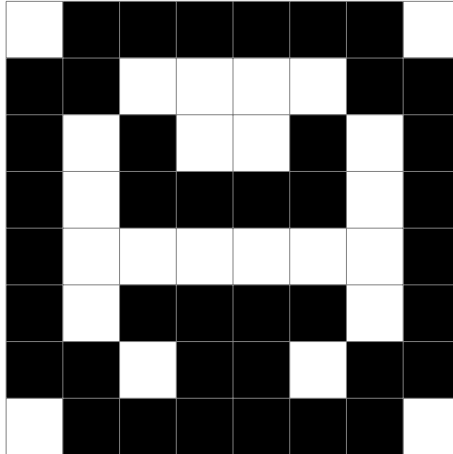
- Arbeitsspeicher für Server (ECC) korrigiert so Bitfehler
- Datenübertragung (→ Kapitel Netzwerke)
- Prüfziffern auf IBAN und Artikelnummern (EAN): gleiche Grundidee



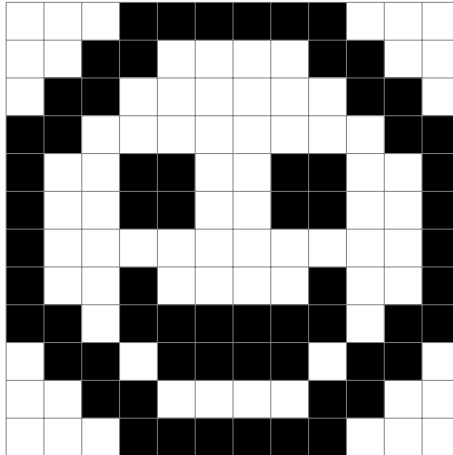
Digitale Bilder



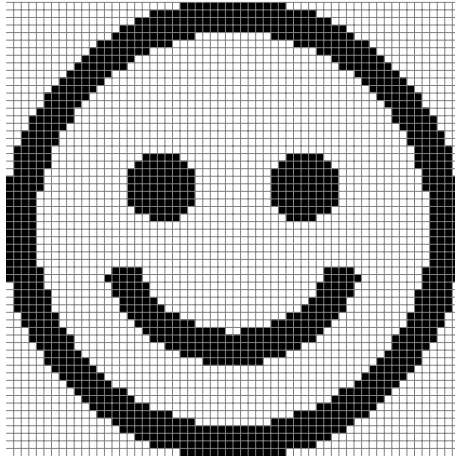




$8 \times 8 = 64$ Bildpunkte



$12 \times 12 = 144$ Bildpunkte



$60 \times 60 = 3600$ Bildpunkte

- Die Auflösung kann beliebig erhöht werden
- Der Speicherbedarf wächst mit steigender Genauigkeit
- Typische Auflösungen:
 - Fernseher (SD): 720 x 576 Bildpunkte
 - Fernseher (UHD): 3.840 x 2.160 Bildpunkte
 - Digitalkamera: min. 60 Bildpunkte je cm
 - Monitor: 1.280 x 1.024 bis 1.920 x 1.080 Bildpunkte

Wie kommen **Farben** in die Bilder?

- Für jedes Pixel wird die Intensität der Grundfarben gespeichert
- Aus diesen wird dann die Farbe gemischt
- RGB
 - Grundfarben: Rot, Grün, Blau
 - Intensitätswerte: 0–255
 - Anzahl der unterschiedlichen Farben: $256^3 = 2^{24} = 16777216$
 - Schreibweise als Hex-Code: **#FF0000** = Rot — drei Byte in Hexadezimal!

Wie kommen **Farben** in die Bilder?

- Für jedes Pixel wird die Intensität der Grundfarben gespeichert
- Aus diesen wird dann die Farbe gemischt
- RGB
 - Grundfarben: Rot, Grün, Blau
 - Intensitätswerte: 0–255
 - Anzahl der unterschiedlichen Farben: $256^3 = 2^{24} = 16777216$
 - Schreibweise als Hex-Code: **#FF0000** = Rot — drei Byte in Hexadezimal!
- Für jede Farbe werden 8 Bit ($2^8 = 256$) benötigt.
- Speicherbedarf ist also 24 mal höher als bei Schwarz/Weiss

Wie kommen **Farben** in die Bilder?

- Für jedes Pixel wird die Intensität der Grundfarben gespeichert
- Aus diesen wird dann die Farbe gemischt
- RGB
 - Grundfarben: Rot, Grün, Blau
 - Intensitätswerte: 0–255
 - Anzahl der unterschiedlichen Farben: $256^3 = 2^{24} = 16777216$
 - Schreibweise als Hex-Code: **#FF0000** = Rot — drei Byte in Hexadezimal!
- Für jede Farbe werden 8 Bit ($2^8 = 256$) benötigt.
- Speicherbedarf ist also 24 mal höher als bei Schwarz/Weiss
- Speicherbedarf eines UHD-Bildes:
 $3.840 * 2.160 * 24 = 199\,065\,600 \text{ bit} = 24\,883\,200 \text{ B} = 24\,300 \text{ kB} = 23,7 \text{ MB}$

Bildbearbeitung

- Größe eines UHD-Bildes: 23,7 MB
- Kamera iPhone MAX: 48 MegaPixel (MP)
- 8064 x 6048 Pixel
- 24 Bit pro Pixel $\approx$ 140 MB Daten!

Bildbearbeitungsprogramme

- Bildbearbeitungsprogramme müssen auf alle Pixel zugreifen!
- Bild muss im Arbeitsspeicher liegen!
- (zusätzlich zu allen anderen Dingen, die der Computer im Arbeitsspeicher braucht)

Werden pixelbasierte Bilder vergrößert, werden die Pixel größer.



Vektorgrafik

- Das Bild wird durch geometrische Figuren beschrieben:
 - Kreis mit Radius 5, Dicke 1 an Position 0, 0
 - gefüllter Kreis mit Radius 1 an Position -1, 1
 - gefüllter Kreis mit Radius 1 an Position 1, 1
 - ...

Vektorgrafiken ...

- können beliebig vergrößert werden, ohne dass Pixel entstehen
- benötigen bei Vergrößerung nicht mehr Speicherplatz

Vektorgrafiken ...

- können beliebig vergrößert werden, ohne dass Pixel entstehen
- benötigen bei Vergrößerung nicht mehr Speicherplatz
- Anwendungen: Diagramme, Firmenlogos, Computerschriften, ...
- Dateiformate: svg, pdf, ...

Vektorgrafiken ...

- können beliebig vergrößert werden, ohne dass Pixel entstehen
- benötigen bei Vergrößerung nicht mehr Speicherplatz
- Anwendungen: Diagramme, Firmenlogos, Computerschriften, ...
- Dateiformate: svg, pdf, ...

<https://de.wikipedia.org/wiki/Vektorgrafik>

Wie werden aus Bildern **Videos**?

- Ab ca. 14–16 Bildern pro Sekunde nimmt das Auge diese als bewegte Szene wahr.
- Filme bestehen aus 24–48 Bildern pro Sekunde.
- Computerspiele verwenden auch 60–390 Bilder pro Sekunde.

Wie werden aus Bildern **Videos**?

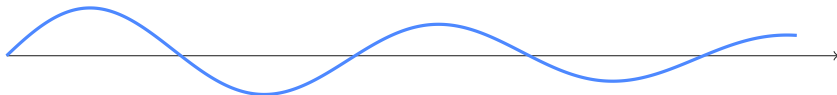
- Ab ca. 14–16 Bildern pro Sekunde nimmt das Auge diese als bewegte Szene wahr.
- Filme bestehen aus 24–48 Bildern pro Sekunde.
- Computerspiele verwenden auch 60–390 Bilder pro Sekunde.
- Jedes Bild benötigt den zuvor genannten Speicherplatz.
- Wieso kann man trotzdem Videos streamen?

Wie werden aus Bildern **Videos**?

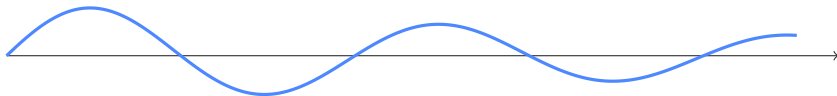
- Ab ca. 14–16 Bildern pro Sekunde nimmt das Auge diese als bewegte Szene wahr.
- Filme bestehen aus 24–48 Bildern pro Sekunde.
- Computerspiele verwenden auch 60–390 Bilder pro Sekunde.
- Jedes Bild benötigt den zuvor genannten Speicherplatz.
- Wieso kann man trotzdem Videos streamen?
- Komprimierungsverfahren reduzieren das Datenvolumen!

CODIERUNG VON TON

- Schall = Schwingung der Luft
- Ein Mikrofon macht daraus eine schwankende **Spannung**

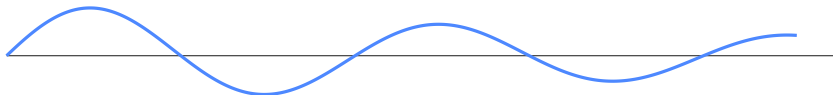


- Schall = Schwingung der Luft
- Ein Mikrofon macht daraus eine schwankende **Spannung**



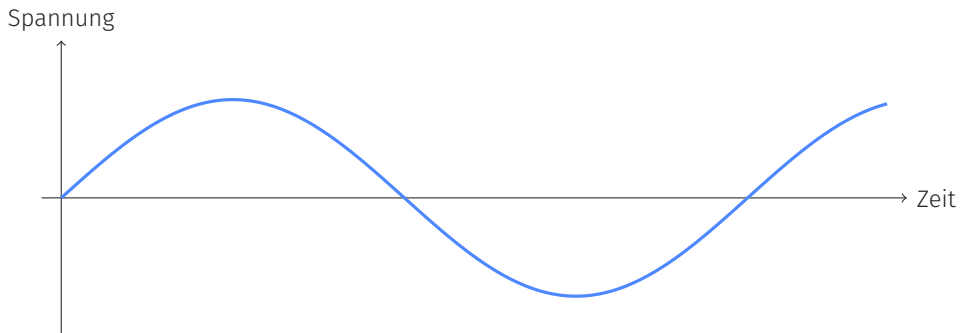
Eine Kurve — wie speichert man die als Zahlen?

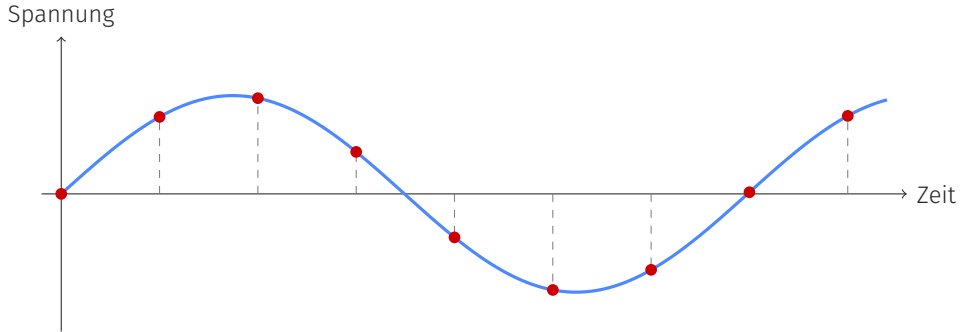
- Schall = Schwingung der Luft
- Ein Mikrofon macht daraus eine schwankende **Spannung**



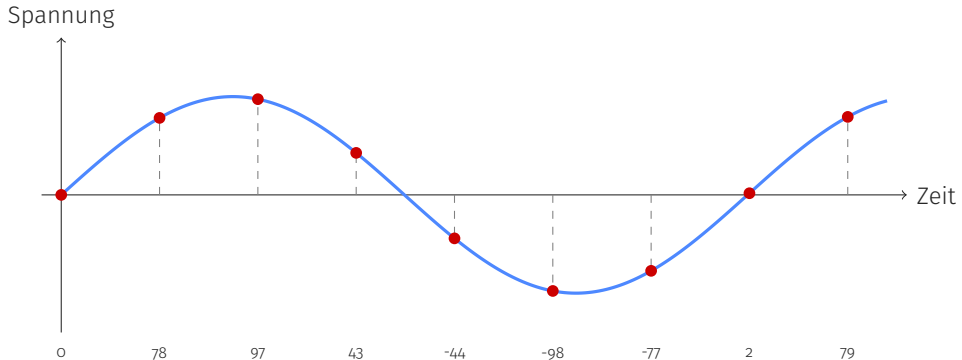
Eine Kurve — wie speichert man die als Zahlen?

- Messen! Genau wie beim Bild: in viele kleine Häppchen zerlegen

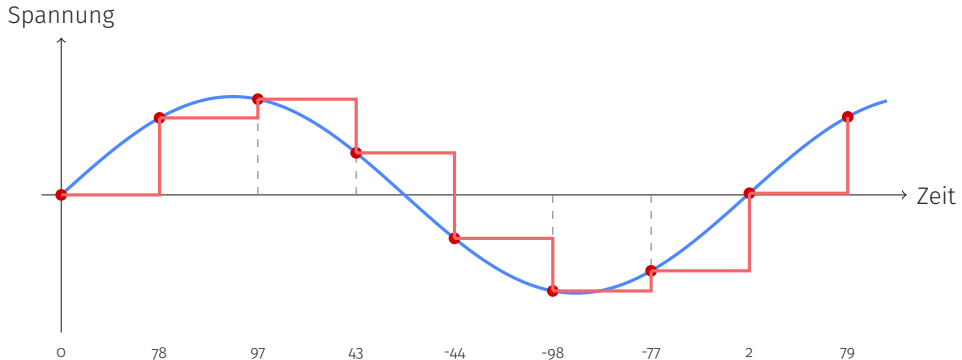




- In festen Zeitabständen den Wert **messen**

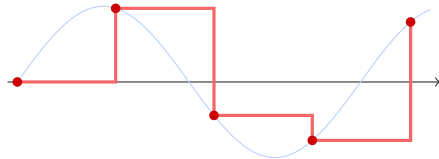


- In festen Zeitabständen den Wert **messen**
- Gespeichert wird nur die Zahlenreihe

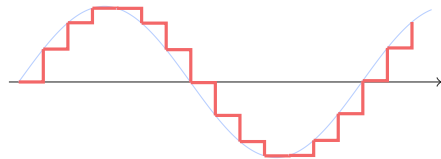


- In festen Zeitabständen den Wert **messen**
- Gespeichert wird nur die Zahlenreihe
- Der Computer kennt nur noch die **Treppe** — die Kurve dazwischen ist weg

Wie oft muss man messen?

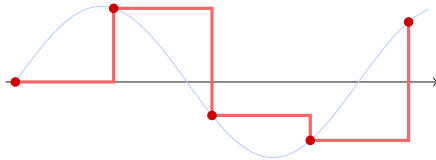


5 Messungen

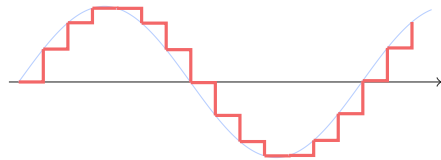


17 Messungen

Wie oft muss man messen?



5 Messungen



17 Messungen

- Zu wenige Messungen: die Kurve geht kaputt
- Viele Messungen: fast das Original — aber mehr Daten
- Wie die Auflösung beim Bild!

Auch jeder **Messwert** selbst wird gerundet — es gibt nur endlich viele Stufen.

- Wenige Stufen: hörbares Rauschen — wie grobe Pixel im Bild
- CD-Qualität: 16 Bit = 65.536 Stufen je Messwert

Auch jeder **Messwert** selbst wird gerundet — es gibt nur endlich viele Stufen.

- Wenige Stufen: hörbares Rauschen — wie grobe Pixel im Bild
- CD-Qualität: 16 Bit = 65.536 Stufen je Messwert
- Das gleiche Prinzip wie bei den Farbstufen:
mehr Bits = feiner = mehr Speicher

Warum misst eine CD ausgerechnet 44.100-mal pro Sekunde?

Warum misst eine CD ausgerechnet 44.100-mal pro Sekunde?

- Menschen hören bis ca. 20.000 Schwingungen pro Sekunde
- Pro Schwingung braucht man gut 2 Messungen
- → etwas mehr als 40.000 Messungen pro Sekunde

Warum misst eine CD ausgerechnet 44.100-mal pro Sekunde?

- Menschen hören bis ca. 20.000 Schwingungen pro Sekunde
- Pro Schwingung braucht man gut 2 Messungen
- → etwas mehr als 40.000 Messungen pro Sekunde

Speicherbedarf

- $44.100 * 16 * 2$ (Stereo) = 1 411 200 bit $\approx$ 1,4 Mbit pro Sekunde
- Rund 10 MB pro Minute Musik

Warum misst eine CD ausgerechnet 44.100-mal pro Sekunde?

- Menschen hören bis ca. 20.000 Schwingungen pro Sekunde
- Pro Schwingung braucht man gut 2 Messungen
- → etwas mehr als 40.000 Messungen pro Sekunde

Speicherbedarf

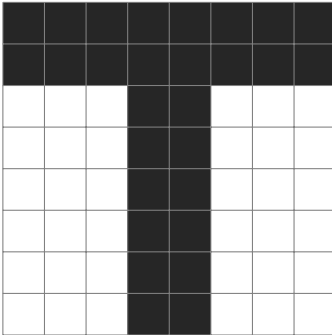
- $44.100 * 16 * 2$ (Stereo) = 1 411 200 bit $\approx$ 1,4 Mbit pro Sekunde
- Rund 10 MB pro Minute Musik
- MP3 braucht ca. 1 MB pro Minute — wie kann das sein?
→ Komprimierung

KOMPRIMIERUNG

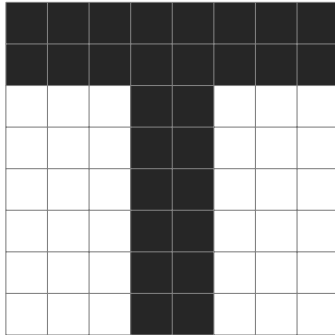
Diktieren Sie AAAAABBB am Telefon — möglichst kurz!

Diktieren Sie AAAAABBB am Telefon — möglichst kurz!

- "5-mal A, 3-mal B" → 5A3B
- Das ist **Lauf längencodierung** (Run-Length Encoding, RLE)
- Nichts geht verloren: aus 5A3B entsteht wieder exakt AAAAABBB
→ **verlustfrei**
- Aus 8 Zeichen werden 4



- 64 Pixel: 0 = weiß, 1 = schwarz
- Schreiben Sie jede Zeile als Zahlenfolge:
erst Anzahl **weiße**, dann **schwarze**,
dann wieder weiße, ...
- Beispiel Zeile 1: komplett schwarz → 0, 8



0, 8

0, 8

3, 2, 3

3, 2, 3

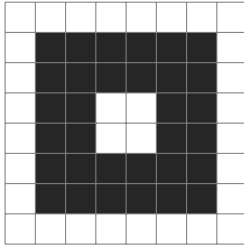
3, 2, 3

3, 2, 3

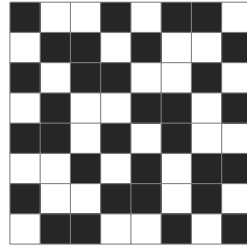
3, 2, 3

3, 2, 3

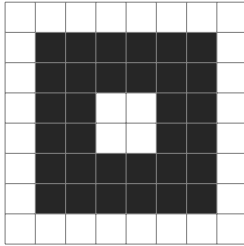
- 64 Pixel: 0 = weiß, 1 = schwarz
- Schreiben Sie jede Zeile als Zahlenfolge:
erst Anzahl **weiße**, dann **schwarze**,
dann wieder weiße, ...
- Beispiel Zeile 1: komplett schwarz → 0, 8
- Statt 64 Bits nur noch 22 Zahlen!



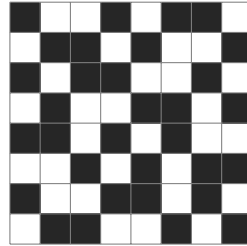
Logo: lange Läufe
→ RLE lohnt sich



Foto/Rauschen: Läufe der Länge 1
→ RLE macht es sogar **größer**



Logo: lange Läufe
→ RLE lohnt sich



Foto/Rauschen: Läufe der Länge 1
→ RLE macht es sogar **größer**

- Logos, Screenshots, Dokumente: verlustfrei sehr wirksam
- Fotos: kaum Gewinn → andere Tricks nötig (JPEG)
- Noch ein Trick: **gleiche Muster** nur einmal speichern — gleich mehr dazu!

Warum sind die Bilder auf dem Speichermedium dann doch nicht so groß?

Komprimierung

- Verlustfrei
 - Wiederkehrendes Auftreten von Bildelementen wird optimiert gespeichert
 - Beispiel-Verfahren: Lauflängencodierung — gerade selbst gemacht!
 - Beispiele: PNG, GIF, BMP
 - Nachteile: insbesondere Fotos werden nur begrenzt kleiner
- Verlustbehaftet
 - Bildelemente, die Menschen kaum wahrnehmen, vereinfacht dargestellt
 - Beispiele: JPEG, MP3
 - Nachteile: insbesondere grafische Bilder können Verfremdungen / Artefakte erhalten

Worst-Case bei Komprimierung

Original		Komprimiert	
110.000	54,60	110.000	54,80
125.000	60,00	125.000	60,00
140.000	65,40	140.000	85,40
155.000	70,80	155.000	70,80
170.000	76,20	170.000	76,20

https://www.dkriesel.com/blog/2013/0802_xerox-workcentres_are_switching_written_numbers_when_scanning

Worst-Case bei Komprimierung

Original		Komprimiert	
110.000	54,60	110.000	54,80
125.000	60,00	125.000	60,00
140.000	65,40	140.000	85,40
155.000	70,80	155.000	70,80
170.000	76,20	170.000	76,20

https://www.dkriesel.com/blog/2013/0802_xerox-workcentres_are_switching_written_numbers_when_scanning

Was war da los?

- Die Scanner-Software speichert ähnliche Bildschnipsel nur **einmal**
- Eine 6 und eine 8 wurden als "gleiches Muster" eingestuft — und vertauscht
- Genau der Trick von eben — nur zu aggressiv eingestellt

Was war da los?

- Die Scanner-Software speichert ähnliche Bildschnipsel nur **einmal**
- Eine 6 und eine 8 wurden als "gleiches Muster" eingestuft — und vertauscht
- Genau der Trick von eben — nur zu aggressiv eingestellt
- Lehre: verlustbehaftete Komprimierung nie für Dokumente mit Zahlen!

Wie werden Videos komprimiert?

- Videos sind ja eine Folge von Einzelbildern
- Zwei aufeinanderfolgende Bilder sind zu einem großen Teil identisch
- Es brauchen nur die Änderungen gespeichert bzw. übertragen werden

ZUSAMMENFASSUNG

WAS BEDEUTET 01000001?

01000001

01000001

- Die Zahl 65?

01000001

- Die Zahl 65?
- Der Buchstabe A?

01000001

- Die Zahl 65?
- Der Buchstabe A?
- Ein sehr dunkles Rot (Rot-Anteil 65 von 255)?

01000001

- Die Zahl 65?
- Der Buchstabe A?
- Ein sehr dunkles Rot (Rot-Anteil 65 von 255)?

Alles richtig!

Bits haben keine Bedeutung —
erst die Interpretation macht aus Daten Informationen.

Alles, was sich zählen oder messen lässt, kann digital gespeichert werden!

- Die Auflösung bestimmt den Speicher- und Rechenbedarf
- Die Verwendung bestimmt die erforderliche Auflösung
- Es gibt Standards, mit denen Informationen digital dargestellt werden
- Hexadezimal: kompakte Schreibweise für Bitgruppen
- Ein Paritätsbit macht Fehler erkennbar
- Komprimierung spart Platz: verlustfrei oder verlustbehaftet